PATENT ABSTRACTS OF JAPAN

(11) Publication number : 07-271829
(43) Date of publication of application : 20.10.1995

(51) Int.Cl. G06F 17/30
G06K 9/00

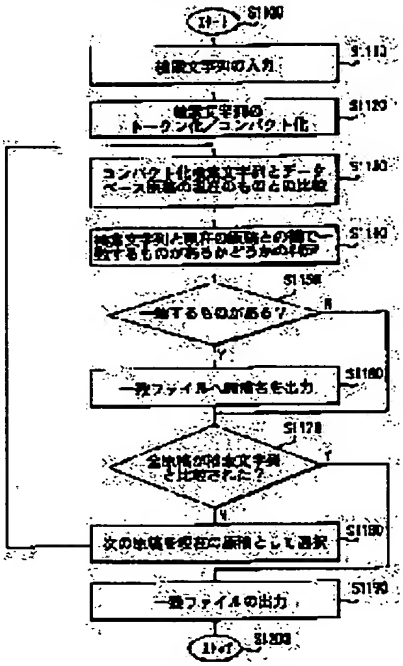
(21) Application number : 07-064381
(22) Date of filing : 23.03.1995

(71) Applicant : XEROX CORP
(72) Inventor : SPITZ A LAWRENCE
DIAS ANTONIO P

(30) Priority
Priority number : 94 220926 Priority date : 31.03.1994 Priority country : US

(54) MATCHING METHOD FOR TEXT IMAGE AND ORIGINAL USING CHARACTER SHAPE CODE

(57) Abstract:
PURPOSE: To accurately match a text image and a text original with each other by a compacted character shape code.
CONSTITUTION: The first method for performing accurate or non-strict matching of an original stored in an original data base includes a processing for converting a data base original into a compacted form made into a token. A retrieval character string or a retrieval original is converted into the compacted form made into the token and compared. Whether or not a test character string is present inside the data base original or whether or not the data base original corresponds to a test original is judged. A second, method for performing the non-strict matching of the test original and the data base original includes the processing for generating one or plural floating point number sets of the respective data base original and test original. The floating point number set of a data base is compared with the floating point number set of the test original and a matching degree is decided. A threshold value is used.



LEGAL STATUS
[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平7－271829

(43)公開日　平成７年(1995)10月20日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　17/30					
Ｇ０６Ｋ　9/00	Ｓ	9289－5Ｌ			
		9194－5Ｌ	Ｇ０６Ｆ　15/ 403	３５０　Ｃ	
		9194－5Ｌ	15/ 40	３７０　Ｂ	

審査請求　未請求　請求項の数１　ＯＬ　（全 11 頁）

(21)出願番号　特願平7－64381

(22)出願日　平成７年(1995)３月23日

(31)優先権主張番号　０８／２２０９２６

(32)優先日　1994年３月31日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　590000798
ゼロックス　コーポレイション
ＸＥＲＯＸ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　ニューヨーク州　14644
ロチェスター　ゼロックス　スクエア
（番地なし）

(72)発明者　エイ　ローレンス　スピッツ
アメリカ合衆国　カリフォルニア州
94303　パロ　アルト　サウザンプトン
ドライヴ　821

(74)代理人　弁理士　中村　稔　（外６名）

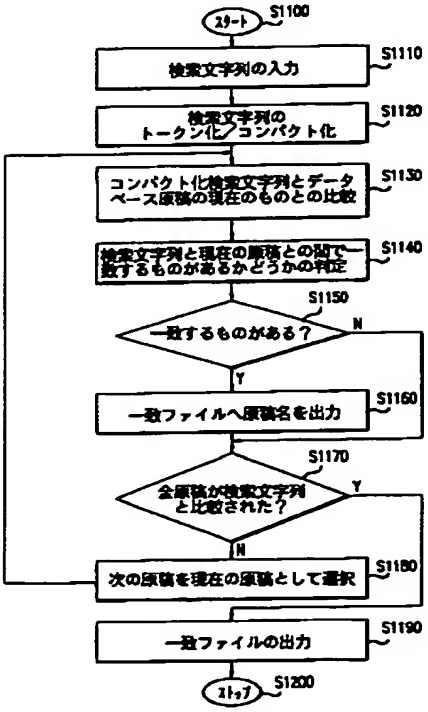
最終頁に続く

(54)【発明の名称】　文字形状コードを用いたテキスト画像と原稿とのマッチング方法

(57)【要約】

【目的】　コンパクト化された文字形状コードで、テキスト画像とテキスト原稿との正確なマッチングを行う。

【構成】　原稿データベースに記憶された原稿の正確または厳密でないマッチングを行う第１の方法は、データベース原稿をコンパクト化されトークン化された形態に変換する処理を含む。検索文字列または検索原稿はコンパクト化トークン化形態に変換され、比較される。テスト文字列がデータベース原稿内にあるかどうか、またはデータベース原稿がテスト原稿と対応するか否かが判断される。テスト原稿とデータベース原稿との厳密でないマッチングを行う第２の方法は、各データベース原稿およびテスト原稿の一または複数の浮動小数点数セットを生成する処理を含む。データベースの浮動小数点数セットはテスト原稿の浮動小数点数セットと比較され、一致度が決定される。しきい値が用いられる。



1

【特許請求の範囲】

【請求項１】　テスト・ストリングを複数の原稿とマッチングするための方法であって、

前記複数の原稿のそれぞれを文字形状コード表記に変換し、

前記複数の原稿のそれぞれに対して、前記文字形状コード表記を、縮小されたバイナリ・データ・ストリングに変換し、

前記複数の原稿のそれぞれの縮小されたバイナリ・データ・ストリングをデータベースに格納し、

テストストリングを入力し、

前記テスト・ストリングを文字形状コード表記に変換し、

前記テスト・ストリングの前記文字形状コード表記を縮小されたバイナリ・データ・ストリングに変換し、

前記テスト・ストリングの前記縮小されたバイナリ・データ・ストリングを前記複数の原稿のそれぞれの縮小されたバイナリ・データ・ストリングとマッチングし、

マッチングした原稿のリストを出力する、

方法。

【発明の詳細な説明】

【０００１】

【技術分野】本発明は、テキスト画像（テキスト・イメージ）を原稿（ドキュメント）と比較するための方法、およびテキスト画像と原稿の文字形状コード表記の比較に基づき、原稿データベース内に原稿を発見するための方法に関する。特に、本発明は、テキスト画像および原稿の文字の全体の物理的形状を表す、制限された個数の文字コードを使用して、テキスト画像と原稿を文字形状コード表記に変換することに関する。

【０００２】

【発明の背景】この技術分野における周知の従来システムにおいては、テキスト画像をテキスト原稿と比較する前に、テキスト画像は、まず、光学式文字読取り（ＯＣＲ：Optical Character Recognition）技術を用いて、分離したテキスト原稿に変換されなければならない。しかしながら、光学式文字読取り過程においては、一般に、認識テキスト原稿内に置換、消滅および挿入のような多くのエラーを有するものが生成される。このように、光学的に認識されたテキスト原稿と電子テキスト原稿との正確な照合を行うことは一般に不可能である。したがって、認識されたテキスト原稿を正確なデータベース・テキスト原稿と正しく照合することは、不可能でない場合であっても、困難であることが多い。

【０００３】

【発明の概要】本発明は、文字形状でコード化されたワード・トークンを含むテキスト画像を、コンパクト化された文字形状コード表記に変換するための方法を提供する。

【０００４】また、本発明は、コンパクト化された文字

2

形状コードで、テキスト画像とテキスト原稿との正確な照合を行う方法を提供する。

【０００５】さらに、本発明は、トークン化されたテキスト画像およびトークン化されたテキスト原稿のそれぞれから生成された複数の浮動小数点数を生成することにより、トークン化されたテキスト画像とテキスト原稿との厳密でない照合を行う方法を提供する。続いて、これらの浮動小数点数は比較され、テキスト画像とテキスト原稿とを厳密でない照合が行われる。

【０００６】本発明のこれらの目的および他の目的を達成するために、第１の好ましい実施例においては、テキスト画像は、まず走査され、続いてトークン化されたテキスト画像に変換される。この変換には、出願係属中の特許出願第6-70296 号（参照のためここに引用）に示されている自動文字形状分類方法が用いられる。

【０００７】電子的に記憶された（すなわち、文字コード・テキストとして記憶された）テキスト原稿のライブラリは、この方法を用いて、トークン化された電子テキスト原稿（以下「トークン化電子テキスト原稿」ともいう）のライブラリに予め変換される。続いて、このトークン化電子テキスト原稿は、本発明のコンパクト化方法の第１の好ましい実施例を用いて、コンパクト化されたトークン化電子テキスト原稿に変換される。同様にして、テキスト画像がトークン化されたテキスト画像（以下「トークン化テキスト画像」ともいう）に変換された後に、これは、本発明のコンパクト化方法の第１の好ましい実施例を用いて、コンパクト化されたトークン化テキスト画像に変換される。続いて、このコンパクト化されたトークン化テキスト画像は、データベースに記憶された原稿のライブラリと比較され、正確に一致（マッチ）するものが決定される。

【０００８】第２の好ましい実施例においては、データベースは、コンパクト化されたトークン化電子テキスト原稿ではなく、コンパクト化されたトークン化テキスト画像のライブラリを含んでいる。電子テキスト文字列（電子テキスト・ストリング）は、コンパクト化され、トークン化された形態に変換され、テキスト画像のデータベース・ライブラリを検索するために使用され、そのテキスト文字列と正確に一致するものを含むテキスト画像が突き止められる。

【０００９】第３の好ましい実施例においては、テキスト画像および電子テキスト原稿は、一または二以上の浮動小数点数からなるセットによってそれぞれ表される。テスト原稿（テキスト画像または電子テキスト原稿のいずれか）の浮動小数点数のセットは、テキスト画像または電子テキスト原稿の複数の他のものを表す浮動小数点のセットのデータベースと比較される。データベース原稿の浮動小数点数のセットは、テスト原稿のセットと比較され、このテスト原稿と一致するテキスト画像または原稿の厳密でない識別が行われる。

【００１０】

【実施例】テキスト画像（テキスト・イメージ）および電子テキスト原稿（電子テキスト・ドキュメント）が以下に示す方法を用いてコンパクト化される前に、このテキスト原稿およびテキスト画像は、まず、電子テキスト原稿の場合には文字コードからトークン化された文字形状コード表記に、テキスト画像の場合にはビットマップからトークン化された文字形状コード表記にそれぞれ変換されなければならない。

【００１１】図１は、文字形状コード認識システムを示している。このシステムは、電荷結合素子（ＣＣＤ）等を有するスキャナ110 を備えている。スキャナ110 は、図２に示すような原稿（ドキュメント）100 を走査する。原稿100 は画像（イメージ）102 を有する。画像102 は、ヨーロッパ・スクリプト・タイプのテキスト文字列を有する。スキャナ110 は、オリジナルの原稿100 の画像102 を含む複数のピクセルの位置および画像濃度を表すディジタル・データ信号を出力する。

【００１２】このディジタル・データ信号はメモリ112 に送られる。メモリ112 では、ディジタル・データ信号が一時的または無期限に記憶される。ディジタル・データ信号は、メモリ112 から出力されると、汎用ディジタル・コンピュータ114 に入力される。コンピュータ114 に入力されると、まず、ディジタル・データ信号は、画像102 の非テキスト部分を除去してテキスト部分104 を残すことにより整理（クリーン・アップ）される。さらに、歪み等のディジタル・データ信号上のスキャナ生成物が補正される。整理されたディジタル・データ信号は、メモリ112 にに再び記憶されるか、または、コンピュータ114 のメモリに記憶される。選択的に、スキャナ110 がスキャナ生成物の除去のような前処理を提供することができる。

【００１３】図１に示すように、本発明の汎用ディジタル・コンピュータ114 は、メモリ22および入出力回路24を備えている。メモリ22には、制御プログラムが記憶される。入出力回路24は、メモリ112 からのディジタル・データ信号を入力し、画像102 のテキスト部分104 または一致（マッチ）する原稿の決定されたワード・トークンを表す信号を出力する。汎用コンピュータ114 は、画像メモリ26、結合コンポーネント生成手段28、空間的特徴決定手段30、文字形状分類手段32、およびコンパクト手段34も備えている。画像メモリ26には、ディジタル・データ信号が記憶される。結合コンポーネント生成手段28は、ディジタル・データ信号から、結合したコンポーネント（結合コンポーネント）を生成する。空間的特徴決定手段30は、テキスト部104 のライン（行）、ワードおよび文字セルの座標、ならびに各文字セル内の各結合コンポーネントの位置を決定する。文字形状分類手段32は、文字セルを、抽象化された文字形状コードに変換する。コンパクト手段34は、決定されたワード・トークン

に基づいて、テキスト部104 を、コンパクト化され、かつ、トークン化された形に変換する。制御プログラムを記憶するメモリ22は、ＲＯＭ22ａまたはＲＡＭ22ｂのいずれを含んでいてもよい。

【００１４】分類手段32の好ましい実施例においては、分類手段32は、結合コンポーネント計数手段320 、トップ位置検出手段322 、ボトム位置検出手段324 、結合コンポーネント・サイジング手段326 、ライン位置記憶手段328 および比較手段330 を備えている。結合コンポーネント計数手段320 は、現在の文字セル内における結合コンポーネントの個数を決定する。トップ位置検出手段322 は、現在の文字セル内の少なくとも一つの結合コンポーネントのトップの位置を突き止める。ボトム位置検出手段324 は、現在の文字セルの少なくとも一つの結合コンポーネントのボトムの位置を突き止める。結合コンポーネント・サイジング手段326 は、現在の文字セルの少なくとも一つの結合コンポーネントの高さと幅を決定する。ライン位置記憶手段328 は、現在の文字セルを含んだラインの基底ラインおよびｘラインの少なくとも一つのものを記憶する。比較手段330 は、結合コンポーネントのトップ位置、結合コンポーネントのボトム位置および結合コンポーネントの高さの少なくとも一つを、基本ラインの位置、ｘラインの位置および結合コンポーネントの幅の少なくとも一つと比較する。

【００１５】コンパクト化を行う方法の第１の好ましい実施例と関連した選択的な実施例においては、前記分類手段はギャップ決定手段332 も含んでいる。ギャップ決定手段332 は、結合コンポーネントの右側部分にギャップが存在するかどうかを判定する。もちろん、結合コンポーネント生成手段28、空間的特徴決定手段30、分類手段32および比較手段34の各機能および対応する手段を、独立した手段によって実現することもできるし、このような構造は、上述した本発明の好ましい実施例に相当することが分かる。コンパレータ36は、コンパクト化されたテキスト文字列または原稿をデータベースと比較する。また、コンパレータ36は、第４の好ましい実施例の浮動小数点の値のセット（集合）も比較する。

【００１６】処理において、図２に示すような画像102 を含む原稿100 は、スキャナ110 の上または内部に置かれ、走査される。そして、シリアルまたはパラレルのディジタル・データ信号が生成される。このディジタル・データ信号は複数の信号部分を含んでいる。各部分はオリジナル画像102 に対応したピクセルを表す。画像102 の各ピクセルは、画像102 における位置および画像濃度を有する。したがって、ディジタル・データ信号の各信号部分は、対応するピクセルの位置および画像濃度を表すデータを含んでいる。

【００１７】続いて、スキャナ110 から出力されたディジタル・データ信号は、メモリ112に記憶される。メモリ112 は、ＲＡＭ、フラッシュ・メモリ、ディスク・メ

モリ等で構成することができる。メモリ112 のタイプに関係なく、ディジタル・データ信号は、各信号部分に含まれる位置および画像濃度データに応じてメモリ112に記憶される。もちろん、ディジタル・データ信号を、この中継を行うメモリ112 に記憶することなく、汎用ディジタル・コンピュータ114 に直接入力することもできる。選択的に、汎用ディジタル・コンピュータ114 にメモリ112 を組み込むこともできる。メモリ112 は、走査された画像102 の長期記憶として使用される場合もある。

【００１８】オペレータがスキャナ110 への原稿の入力を完了するか、またはそうでなれば、システムが、画像102 を表すディジタル・データ信号を文字形状コード・シンボルに変換すべきと判断すると、画像102 を表すディジタル・データ信号はメモリ112 から汎用コンピュータ114 へ出力される。もちろん、特殊な用途のディジタル・コンピュータまたはハードウェア・ロジック回路を、汎用ディジタル・コンピュータ114 に代わって使用することもできる。

【００１９】メモリ112 に記憶されたディジタル画像データ信号は、汎用コンピュータ114に出力され、入出力手段24を介して画像メモリ26に入力される。ディジタル・データ信号の画像メモリ26への記憶が完了すると、続いて、結合コンポーネント生成手段28がディジタル・データ信号を処理する。結合コンポーネント生成手段28は、画像102 を表すディジタル・データ信号を複数の結合コンポーネントに分解する。各結合コンポーネントは、単一のラインの一または複数の信号部分を備えている。各結合コンポーネントは、オリジナル画像102 におけるある最小の画像濃度を有し、かつ、連続した経路を形成するピクセルに対応した信号部分を含んでいる。一般に、各スクリプト文字は“Ｆｕｊｉ”の“Ｆ”のように一つの結合コンポーネントに対応するものもあり、また、“Ｆｕｊｉ”の“ｊ”または“ｉ”のように二以上の結合コンポーネントに対応するものもある。

【００２０】結合コンポーネント生成手段28が、画像102 の複数の結合コンポーネントをディジタル・データ信号から生成すると、画像102 に対応したディジタル・データ信号、ならびに結合コンポーネント生成手段28によって生成された結合コンポーネントのリストおよびそれらの位置は、画像メモリ26に記憶され、空間的特徴決定手段30に出力される。

【００２１】空間的特徴決定手段30は、ラインの位置、ワードのスペースおよび文字セルのようなテキスト部分の空間的特徴を決定する。各文字セルは、隣接するスペース間のライン内の垂直方向に並んだ結合コンポーネントを含んでいる。例えば、“Ｆｕｊｉ”の文字“ｉ”および“ｊ”は、２つの独立した結合コンポーネントからそれぞれ形成されている。空間的特徴決定手段30は、一ラインの垂直方向に並んだ全ての結合コンポーネントを一つの文字セルにグループ化する。結合コンポーネントの生成および結合コンポーネントからテキスト部分104 の空間的特徴の決定を行う一つの好ましい方法および装置は、米国特許出願第07/047,514号に示されている。この米国出願は、本特許出願と同一の出願人により出願されたものであり、参照のためにここに引用される。

【００２２】続いて、結合コンポーネントおよび文字セルのリストは、空間的特徴決定手段30によって出力され、文字形状コード分類手段32に与えられる。文字形状コード分類手段32は、文字セル内の結合コンポーネントまたは結合コンポーネント群を、文字セル内の結合コンポーネントの個数および位置に基づいて複数の抽象的な文字コード（抽象文字コード）の一つに変換する。図３は、好ましい文字コード・リストおよび各コードに対応する文字を示している。図３に示すように、13個の異なった抽象文字形状コードが使用される。しかしながら、第１および第２の好ましい実施例においては、これらの13個の文字形状コードはコード“Ａ，Ｕ，ｉ，ｘ，ｇ，ｊ”に限定される。さらに、“スペース”および“ＣＲ”の２つの追加コードが使用される。コード“スペース”はワード間のスペースを示す。コード“ＣＲ”（キャリッジ・リターン）はラインの終了を示す。各抽象文字形状コードは、文字セル内の独立した結合コンポーネントの個数、各文字セルの独立した結合コンポーネント間の相対的な位置、および文字セル内の結合コンポーネントの位置に基づいて、一または複数の文字を代表するものである。

【００２３】上述したトークン化を行うシステムの処理を簡単化したフローチャートが図４に示されている。ステップS100において、システムは処理を開始する。原稿がステップS110で走査され、ディジタル・データ信号が生成される。続いて、ステップS120において、ディジタル画像データ信号は整理（クリーンアップ）される。この整理は、所望の前処理アルゴリズムをこのディジタル画像データに適用することによって行われる。ステップS130において、ディジタル画像データ信号の結合コンポーネントが識別され、ステップS140において、文字セルが決定される。ステップS150において、各文字セルの文字タイプ分類が決定される。ステップS160において、文字コードがともにグループ化され、ワード間およびワードの内部スペースに基づいてトークンが形成される。

【００２４】図３に示すコード化を実行する決定木が図５に示されている。図５に示すように、１つの結合コンポーネントを有する文字セル用の７つの抽象文字コード、２つの結合コンポーネントを有する文字セル用の５つの抽象文字コード、および３つの結合コンポーネントを有する文字セル用の１つの抽象文字コードがある。

【００２５】本発明の方法の好ましい実施例は図５に示す決定木を実行する。ステップS300において、分類手段32は、まず、現在の文字セル内の結合コンポーネントの

個数を決定する。本発明の好ましい実施例において、文字形状コード分類手段32は、テキスト部分104 の各文字セルをセルごとに処理する。

【００２６】本発明の方法および装置は統計的に強固な全体の特徴を分析するので、この方法および装置は非常に質の悪い印刷および（または）走査された原稿であっても処理でき、歪みが容易に生じる従来のＯＣＲ技術と対べて、優れた特徴を分析できる。したがって、ディジタル・データ信号またはこのディジタル・データ信号から生成された結合コンポーネントに対して、原稿の全ての文字を完全に表すことは必要ない。むしろ、本発明はよく起こる走査エラーに耐えることができる。すなわち、本発明は、例えば、１つの結合コンポーネント文字が２つまたはそれ以上の結合コンポーネントに分離したり、２つまたはそれ以上の分離した結合コンポーネントが１つの結合コンポーネントに合体したり、結合コンポーネントがライン上に間違って置かれたりするようなエラーに対して強い。また、本発明の方法および装置は、歪みおよび（または）ねじれを有する画像を分析するときにも強さを発揮する。

【００２７】ステップS300において、分類手段32が、文字セルが一つだけの結合コンポーネントを有すると判断すると、次に、分類手段32は、その結合コンポーネントのトップ位置が現在のラインのｘライン（ミーンライン）位置より上にあり、かつ、ボトム位置がベースライン（並び線）より上にあるかどうかを判定する。ライン位置および結合コンポーネントの位置は、最上部の位置またはその近傍および最左端の位置またはその近傍にある基準位置から測定され、下方および右方がそれぞれ正になるように測定されることが分かる。

【００２８】ステップS310が肯定的ならば、分類手段32は、ステップS320において文字セルをアポストロフィ（省略符号）に変換する。一方、ステップS310が否定的ならば、分類手段32はステップS330に進む。ステップS330において、分類手段32は、結合コンポーネントのトップ位置がｘライン位置より上にあり、かつ、結合コンポーネントのボトムがベースライン位置上またはそれより下にあるかどうかを判定する。ステップS330が肯定的ならば、分類手段32はステップS340において文字セルを“Ａ”に変換する。“Ａ”は、図３に示すように、全ての大文字、全ての数字、アセンダを有する小文字、および一般に垂直方向（縦方向）を向いた句読点（パンクチュエーション・マーク）の全てを代表するものである。

【００２９】ステップS330が否定的ならば、分類手段32はステップS350に進む。ステップS350において、分類手段32は、結合コンポーネントのトップがｘライン位置より下にあり、かつ、結合コンポーネントのボトムがベースライン位置よりも上にあるかどうかを判定する。ステップS350が肯定的ならば、分類手段32はステップS360において文字セルをハイフンに変換する。

【００３０】ステップS350が否定的ならば、分類手段32はステップS370に進む。ステップS370において、分類手段32は、結合コンポーネントのトップ位置がｘライン位置より下にあり、かつ、結合コンポーネントのボトム位置がベースライン位置より下にあるかどうかを判定する。ステップS370が肯定的ならば、分類手段32は、ステップS380において文字セルをカンマに変換する。ステップS370が否定的ならば、分類手段32はステップS390に進む。ステップS390において、分類手段32は、結合コンポーネントのトップ位置がｘライン位置の下にあるかどうかを判定する。ステップS390が肯定的ならば、分類手段32はステップS400において文字セルをピリオドに変換する。

【００３１】ステップS390が否定的ならば、分類手段32はステップS410に進む。ステップS410において、分類手段32は、結合コンポーネントのボトム位置がベースライン位置より下にあるかどうかを判定する。ステップS410が肯定的ならば、分類手段32はステップS420でその文字セルを“ｇ”に変換する。コード“ｇ”は、図３に示すように、ディセンダを有する任意の小文字を代表するものである。

【００３２】ステップS410が否定的ならば、分類手段32はステップS430に進む。ステップS430において、分類手段32は結合コンポーネントがアセンダまたはディセンダのいずれも有しない小文字であると仮定し、その結合コンポーネントを“ｘ”に変換する。続いて、ステップS430、またはステップS320、S340、S360、S380、S400およびS420に続いて、分類手段32は次の文字セルを現在の文字セルとして選択して、ステップS300に戻る。

【００３３】一方、ステップS300において、分類手段32が、現在の文字セル内に２つの結合コンポーネントがあると判断すると、分類手段32はステップS440に進む。ステップS440において、分類手段32は上部にある結合コンポーネントの高さが上部にある結合コンポーネントの幅の３倍よりも大きいかどうかを判定する。結合コンポーネントの高さとは、そのトップ位置とボトム位置との間の差であり、結合コンポーネントの幅はその右端位置と左端位置との間の差である。ステップS440が肯定的ならば、分類手段32はステップS450に進む。ステップS450において、分類手段32はその文字セルを感嘆符（！）に変換する。

【００３４】ステップS440が否定的ならば、分類手段32はステップS460に進む。ステップS460において、分類手段32は上部にある結合コンポーネントのトップ位置がｘライン位置より上にあり、かつ、下部にある結合コンポーネントのボトム位置がベースライン位置より下にあるかどうかを判定する。ステップS460が肯定的ならば、分類手段32はステップS470においてその文字セルを“ｊ”に変換する。“ｊ”は、ｘラインよりも上に延びる分離した結合コンポーネントおよびベースラインよりも下に

延びる分離した結合コンポーネントを有する任意の小文字を代表するものである。

【００３５】ステップS460が否定的ならば、分類手段32はステップS480に進む。ステップS480において、分類手段32は、上部にある結合コンポーネントのトップ位置がｘライン位置より上にあり、かつ、ボトム位置がベースライン位置より下にないかどうかを判定する。ステップS480が肯定的ならば、分類手段32はステップS490においてその文字セルを“ｉ”に変換する。“ｉ”は、図３に示すように、ｘライン位置より上に延びる分離した結合コンポーネントと、ベースライン位置より下には延びていない分離した結合コンポーネントとを有する任意の小文字を代表するものである。

【００３６】ステップS480が否定的ならば、分類手段32はステップS500に進む。ステップS500において、分類手段32は、上部および下部にある結合コンポーネントの双方がそれらの高さの３倍の幅を有するかどうかを判定する。ステップS500が肯定的ならば、分類手段32はステップS510においてその文字セルを“＝”に変換する。ステップS500が否定的ならば、分類手段32はその文字セルが“：”に変換されるべきと仮定して、ステップS520において、その文字セルはそのように変換される。ステップS520、ならびにステップS450、ステップS470、ステップS490およびステップS510の後、分類手段32は次の文字セルを現在の文字セルとして選択して、ステップS300に進む。

【００３７】一方、分類手段32が、ステップS300において、現在の文字セル内に３つの結合コンポーネントがあると判断すると、分類手段32はステップS530に進む。ステップS530において、分類手段32は、文字セルがウムラウト符号を有する大文字または小文字を表すものと仮定して、これにより、図５に示すように、その文字セルを“U”に変換する。続いて、分類手段32は、次の文字セルを現在の文字セルとして選択し、ステップS300に進む。一方、次の文字セルがないならば、分類手段32はテキスト部分104 の分類を終了し、文字セルの代わりに抽象文字コードのリストを画像メモリ26に出力する。このようにして、図２のテキスト画像は、図６に示す文字形状コード表記に変換される。

【００３８】一または二以上のコード化された文字のワード・トークン・リストは、コンパクト化手段34に与えられる。コンパクト化手段34は、テキスト部分104 を表すワード・トークンのリストを入力し、続いて“Ａ、Ｕ、Ｉ、Ｘ、Ｇ”を除くトークンの全てをテキスト画像から消去して、コンパクト化され、かつ、トークン化されたテキスト部分104 の形態を生成する。もちろん、ＡＳＣＩＩ（アスキー）のような文字コード体系ですでに表されている電子テキスト原稿の場合には、コントローラ114 の文字コード変換器36は、一般に、ＡＳＣＩＩコードを文字形状コードに直接変換することができる。

【００３９】トークン化されたテキスト画像（以下「トークン化テキスト画像」ともいう）およびトークン化された電子テキスト原稿（以下「トークン化電子テキスト原稿」ともいう）が生成されると、これらのトークン化テキスト画像およびトークン化電子テキスト原稿は、コンパクト化手段34によって、コンパクト化されたトークン化テキスト画像および原稿に変換される。この変換において、図７に示すように、コンパクト化方法の第１の好ましい実施例が用いられる。第１の好ましい実施例において、ステップS1000 で、変換プロセスがスタートする。ステップS1010 において、トークン化テキスト原稿は文字形状コードのフル・セットから文字形状コードの縮小セットに縮小される。これらの縮小された文字形状コード（Ａ、Ｕ、ｇ、ｉ、ｊ、ｘ、［スペース］および［CR］）は、実際の原稿に使用されるワード・トークンの大多数に使用されるコードを表す。可能な文字形状コードをこれらの８つの縮小された文字形状コードに限定することにより、文字形状コードを、８つの異なる３ビットのバイナリ数に変換することができる。すなわち、例えば、文字形状コード“Ａ”はバイナリ数“０００”に変換され、“Ｕ”は“００１”に変換され、文字形状コード“ｇ”は“０１０”に変換される。この限定セットの他の文字形状コードは、同様にして、ユニークな３ビットのバイナリ数に変換される。トークン化テキスト原稿が、ステップS1010 において、文字形状コードの限定セットからバイナリ数の列に変換されると、このビット列は、ステップS1020 において、バイトの境界に渡って覆う３ビットのバイナリ・コードを必要なものとして有する、８ビットのバイトにグループ化される。すなわち、（２＋2/3 ）個の形状コードが各８ビットのバイトに圧縮される。続いて、ステップS1030 において、圧縮された「コンパクト化されたトークン化」コードは、［コンパクト化コードのバイト数］［コンパクト化コードのバイト］の形で記憶される。続いて、処理はステップS1040 で終了する。

【００４０】図８は、検索文字列（検索ストリング）を使用してコンパクト化されたトークン化テキスト・ファイルを検索する方法の第１の好ましい実施例を示している。ステップS1100 からスタートし、ステップS1110 において、検索文字列が入力される。ステップS1120 において、検索文字列は、ステップS310～S530およびステップS1000 ～S1040 の方法を使用して、まず、トークン化され、続いて、コンパクト化される。次に、ステップS1130 において、コンパクト化されたトークン化検索文字列は、コンパクト化されたトークン化データベース内において、第１のコンパクト化されたトークン化原稿と比較され、検索文字列と検索している原稿の一部との間で正確に一致するものが突き止められる。ステップS1140 において、システムは、正確に一致するものがコンパクト化されたトークン化検索文字列とコンパクト化された

トークン化テキスト原稿との間に存在するかどうかを判定する。ステップS1150 において、正確に一致するものがステップS1140 で突き止められると、制御はステップS1160 に進む。ステップS1160 で、一致ファイルと正確に一致するものを有するコンパクト化されたトークン化テキスト原稿の名前が出力される。続いて、制御はステップS1170 に進む。一方、ステップS1150 において、正確に一致するものがステップS1140 で発見されなかったならば、制御はステップS1170 に直接進む。

【００４１】ステップS1170 において、システムはデータベースの全てのエントリが分析されたかどうかを判定する。そうでなければ、制御はステップS1180 に進む。ステップS1180 において、次の原稿が選択される。続いて、制御はステップS1130 に戻る。データベース内の全ての原稿が検索されると、制御はステップS1190 に進み、一致するファイルを出力し、テキスト文字列と一致したデータベース・ファイルをリストする。続いて、制御はステップS1200 に進み、処理は終了する。

【００４２】このようにして、第１の好ましい実施例（正確に一致するものを発見する）においては、テスト文字列は、直接入力されるか、またはテキスト画像から取り出される。テスト文字列は、トークン化され、かつ、コンパクト化される。続いて、テスト文字列は、コントローラ114 のコンパレータ36を用いて、コンパクト化されたトークン化テキスト画像またはコンパクト化されたトークン化電子テスト原稿の各エントリとそれぞれ比較され、正確に一致するものが突き止められる。正確に一致するものを有する各テキスト原稿は、テキスト文字列を有するデータベースの各原稿を識別するために出力される。

【００４３】処理において、コンパクト化されたトークン化テスト文字列は、コンパクト化されたトークン化原稿または画像と、左から右へビットごとに、各原稿のビット列の開始部からビット列の終了部に向けて比較される。

【００４４】第１の好ましい実施例の第１の変形例においては、ステップS1110 ～S1140 でテスト文字列の入力および比較を行うのではなく、検索原稿全体が使用される。したがって、検索原稿とデータベースの複数の原稿のそれぞれとの比較の準備のために、原稿全体がトークン化され、かつ、コンパクト化される。

【００４５】第１の好ましい実施例の第２の変形例においては、分類手段32は、最初に結合コンポーネントをステップS430において“ｘ”の文字形状コード・シンボルに変換した後に、結合コンポーネントの右側の中央部分がベースライン位置とｘラインの位置との間で開いた部分を有するかどうかを判定する。そのような開きがある場合には、文字分類手段34は文字形状コード“ｘ”を文字形状コード“ｅ”に変換する。文字形状コード“ｅ”は、文字“ｅ”および“ｃ”を代表するものである。この第１の実施例の第２の変形例において、トークン化画像またはトークン化原稿がステップS1010 で縮小されると、限定されたコード・セットは文字形状コード［CR］ではなくむしろ文字形状コード“ｅ”を含むものとなる。一方、この第１の実施例の第２の変形例では、一致処理は、比較ステップの実行のために、第１および第２の好ましい実施例のいずれも使用することができる。

【００４６】第２の好ましい実施例において、電子検索文字列は、トークン化されたテキスト画像のデータベースを検索するために使用される。

【００４７】第３の好ましい実施例において、コンパクト化手段34は、ステップS1000 ～S1040 のように、トークン化原稿を縮小トークン化原稿に変換して、この原稿をコンパクト化する代わりに、縮小トークン化原稿を使用して、少なくとも一つの浮動小数点数のセット（集合）を生成する。表１に示すように、少なくとも５つの可能な浮動小数点数を、値１～５として識別し、生成することができる。

【００４８】

【表１】

値	定　義
1	素数をマップされた連続した形状コードの全てのペアをXOR 演算し、その結果を加算したもの
2	「平均形状コード」（各文字セルの形状コードの素数値の合計を形状コードの個数で割ったもの）
3	マップされた素数によって乗算されたときの各形状コードの発生の合計
4	連続したコードのペアをXOR 演算する代わりに、全ての他のコードがXOR 演算される点を除いて、値１と同じもの
5	ワード境界がクロスしない点を除いて値１と同じもの。すなわち、所与“of the” map[o]＾map[0], map[f]＾map[o], map[t]＾map[o],map[h]＾map[t], map[e]＾map[h]のXOR

【００４９】表１に示すように、第１の値は、各文字形状コード・シンボルを数字のセットの一つに変換することにより生成される。数字のセットを構成する数字は、すべて相対的に互いに素である。続いて、連続した形状コードの各ペアにXOR 演算が施される。各XOR 演算の結果は加算され、第１の値が生成される。

【００５０】第２の値は、トークン化ファイルの文字を表す数字を加算して、その合計値をテキスト文字列または原稿のトークン化文字の全体の個数によって割ることにより生成される。第３の値は、各文字形状コードがテキスト文字列または原稿に現れる回数をそれを表す数字に乗じ、続いて各文字コードの乗算結果を加算することにより形成される。

【００５１】第４の値は、各隣接したペアではなく、全ての他のコードがXOR 演算される点を除いて、第１の値と同様にして生成される。最後に、第５の値は第１の値と同様にして生成される。しかしながら、第５の値において、XOR 演算は、ワードの境界に渡って拡張されない（各ワード・トークンのリーディング・スペースを含むが）。

【００５２】データベースの各テキスト画像または原稿に対するこれらの値の少なくとも一つと、検索文字列または原稿に対する対応する値とを生成することによって、テキスト文字列または原稿とデータベースを形成する画像またはテキスト原稿との間の正確でない（厳密でない）一致を突き止めることができる。すなわち、図９に示すように、ステップS2000 でスタートした後、テスト原稿の少なくとも一つの浮動小数点数のセットがステップS2010 で生成される。続いて、ステップS2020 において、テスト原稿の浮動小数点数のセットのそれぞれ一つの値は、原稿データベースのテキスト画像またはテキスト原稿の現在のものの浮動小数点数のセットのそれぞれ一つの対応する浮動小数点数と比較される。続いて、ステップS2030 において、距離メジャーが、原稿データベースのエントリの現在のもの用に生成される。この距離メジャーは、テスト原稿の浮動小数点の値のセットと原稿データベースの現在のエントリの浮動小数点の値のセットとの間の全体の相違および類似を表す。

【００５３】続いて、この距離の値はステップS2040 で分析され、テスト原稿と現在のエントリとの間の少なくとも厳密でない一致点を示しているかどうかが判定される。ステップS2050 で、少なくとも厳密でない一致点が発見されると、ステップS2060 において、一致する原稿の名前が一致ファイルへ出力される。一致するものが発見されないならば、制御はステップS2050 から直接ステップS2070 に進む。続いて、ステップS2070 において、データベースはチェックされ、全てのエントリがテスト原稿と比較されたかどうかが判断される。そうでなければ、制御はステップS2080 に進む。ステップS2080 において、次のエントリが選択され、制御はステップS2030 に戻る。そうであれば、少なくとも厳密でない一致するエントリのリストがステップS2090 で出力される。ステップS2100 でシステムは停止する。検索文字列またはデータベースとテキスト画像またはテキスト原稿との間の少なくとも厳密でない一致点として識別されたものが正当であることを確認するために、しきい値が用いられる。これに加えて、オペレータがテキスト画像またはテキスト原稿と検索原稿とをどの程度の近似度で一致させたいかに依存して、このしきい値は調整可能である。

【００５４】上述した第１の好ましい実施例を用いて、英語、フランス語およびドイツ語の原稿を有する多重言語原稿データベースが、これらの言語の話し手に興味の

15

ある167 個の異なるテスト文字列を用いて検索された。UNIXの“AGREP ”コマンド（近似した文字列の一致検索を行う）を用いて、文字列のコンパクト化され、トークン化された形態のものがデータベースの原稿のコンパクト化され、トークン化された形態のものと比較された。167 個の検索文字列の68％（114 個の検索文字列）が、フォールス・ポジティブが返されず、すなわちフォールス・ネガティブが返された。他の32％（54個の検索文字列）は、エラー・レートが増加した。しかしながら、これらの54個の検索文字列は、より短く、あまり特異性のない文字列になる傾向があった。

【００５５】第２の好ましい実施例において、46個の英語、フランス語およびドイツ語の原稿がトークン化され、かつ、コンパクト化されて、データベースを形成している。これらの原稿のうちの36個の走査されたものが、走査された原稿をトークン化するが、データベースの原稿をコンパクト化しないことにより、オリジナル原稿と比較された。続いて、走査された原稿の縮小トークン化形態が、オリジナル原稿の縮小トークン化形態と比較された。36個のオリジナル原稿を生成するときに、特別な注意は払われなかった。したがって、これらの原稿は1.8 ° 上向きにスキューしていた。このテストにおいて、36個の全原稿はそのオリジナルと正確に一致した。さらなるテストにおいて、写真複写の複数の生成物が走査され、データベース原稿と比較された。この第２の好ましい実施例は、３番目および４番目の写真複写生成物にまで耐えた。さらに、この第２の好ましい実施例は、場合によっては、７番目の写真複写生成物でさえも正確に一致させることができた。

【００５６】これらの上記テストは300dpiの走査解像度で行われた。さらなるテストは、200dpiのテキスト画像を走査することにより行われた。このより低い解像度のテストにおいて、36個の原稿のうちの34個の原稿が、データベースのオリジナルの原稿と正確に一致した。２つの原稿がデータベースの原稿のいずれにも一致しなかったが、フォールス・ポジティブは生成されなかった。

【００５７】最後のテストは浮動小数点のフルの５次元

16

セットを使用し、これらの36個の走査された原稿がデータベースの原稿と比較された。第４の好ましい実施例が最も近似して一致するものを発見するために試験され、一致度95％のしきい値が使用された。この第４の好ましい実施例を用いて、36個の走査されたテキスト画像のうちの34個の対応するオリジナル原稿が最も近似して一致するものとして識別された。他の２つのものでは、対応するオリジナル原稿は２番目に近似して一致するものであった。

【図面の簡単な説明】

【図１】文字形状コード認識システムのブロック図である。

【図２】オリジナルの原稿を示す。

【図３】文字形状コードのシンボルおよび実際の対応するスクリプト文字を示す。

【図４】本発明による文字形状コード分類方法の好ましい実施例のフローチャートを示す。

【図５】文字形状コード分類の決定木を示す。

【図６】図３の文字形状コード分類計画によって変換された図２のテキスト部分を示す。

【図７】コンパクト化方法の第１の好ましい実施例のフローチャートを示す。

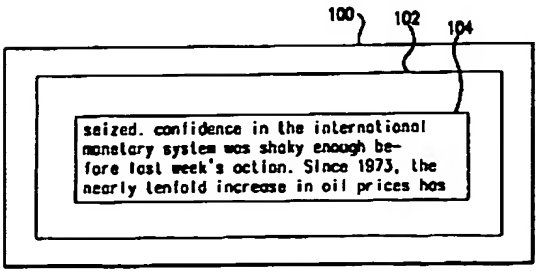
【図８】テキスト・ファイルの検索およびマッチング方法の第１の好ましい実施例のフローチャートを示す。

【図９】テキスト・ファイルの検索およびマッチング方法の第４の好ましい実施例のフローチャートを示す。

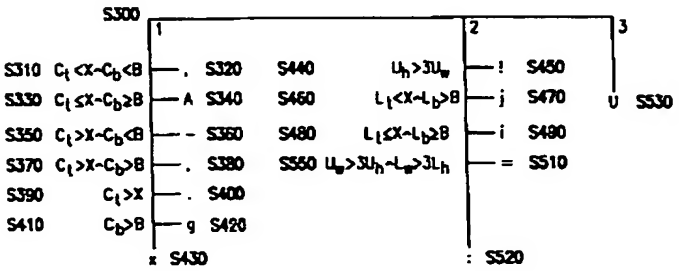
【符号の説明】

110 スキャナ
112 メモリ
114 コントローラ
24　入出力装置
26　画像メモリ
28　結合コンポーネント生成器
30　空間的特徴決定手段
32　文字タイプ分類手段
34　コンパクト化手段
36　比較器

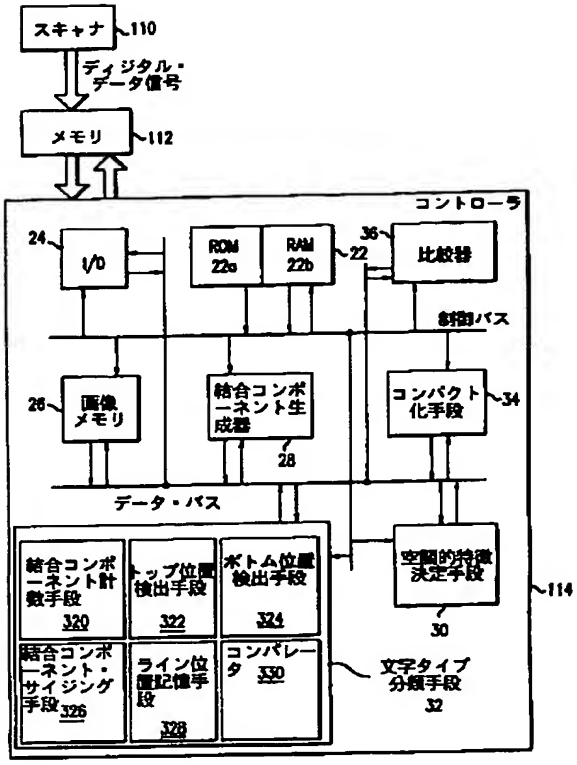
【図２】



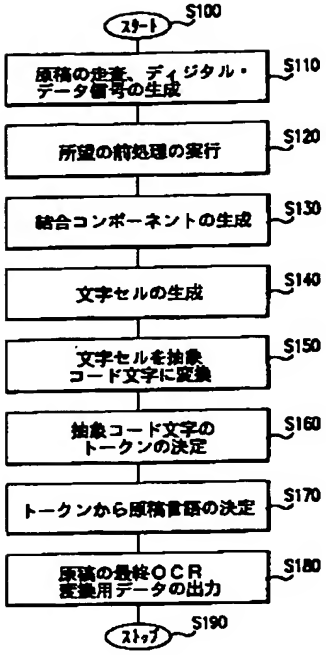
【図５】



【図１】



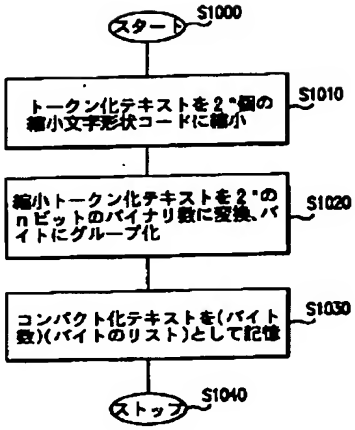
【図４】



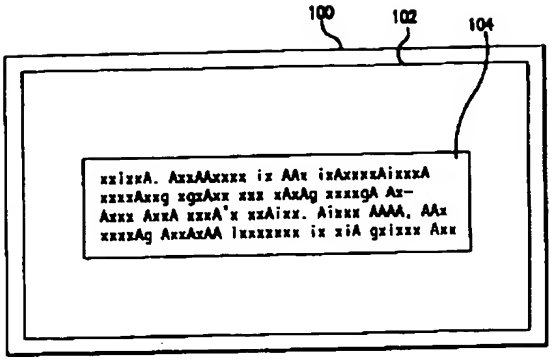
【図３】

文字形状コード	メンバ
A	ABCDEFGHIJKLMNOPQRSTUVWXYZ
x	
i	
6	
j	j
'	'
-	
,	,
.	.
:	
=	
U	
l	l

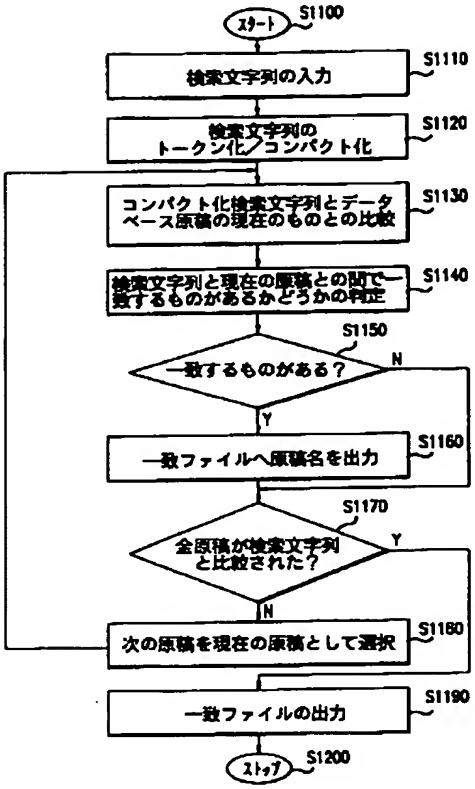
【図７】



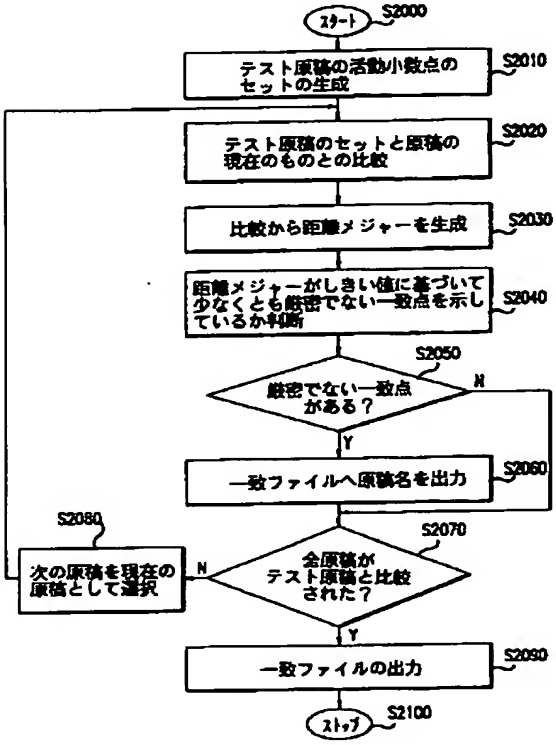
【図６】



【図８】



【図９】



フロントページの続き

(72)発明者　アントニオ　ピー　ディアス
アメリカ合衆国　マサチューセッツ州　02138ケンブリッジ　ハーヴァード　ユニヴァーシティー　クインシー　メイル　センター　200